AMENDMENTS TO THE SPECIFICATION

In the Specification:

Please amend the paragraph beginning on page 1, line 5 starting with "The present invention" with the following amended paragraph:

The present invention relates generally to computer system(s), and, more particularly, to a system and method for detecting direct memory access (DMA)-generated memory corruption in transaction-based DMA bus system(s) (*e.g.,* a PCI express® peripheral component interconnect ~~interface~~ express bus system).

Please amend the paragraph beginning on page 2, line 20 starting with "The present invention" with the following amended paragraph:

The present invention provides for a system and method that facilitate detection of direct memory access (DMA) corruption. The system can mitigate DMA memory corruption in transaction-based DMA bus system(s) (*e.g.,* PCI Express® peripheral component interconnect ~~interface~~ express) by rejecting disallowed transaction(s). In accordance with an aspect of the present invention, the system is extended to include an interface to specify "allowed" and/or "disallowed" DMA transactions. If a disallowed DMA transaction occurs, then it is rejected and, optionally, an error is raised. For example, the determination of whether a transaction can be allowed can be based on a source identifier and/or memory address(es) involved.

Please amend the paragraph beginning on page 2, line 29 starting with "Thus, the system" with the following amended paragraph:

Thus, the system of the present invention can facilitate detection of direct memory access transaction(s) that can, if permitted, cause memory corruption. The system can further facilitate identification (*e.g.,* to the operating system) of the direct cause of the transaction(s) (*e.g.,* PCI

Express® peripheral component interconnect express source identifier) that, if permitted, can cause memory corruption.

Please amend the paragraph beginning on page 3, line 17 starting with "In one example" with the following amended paragraph:

In one example, PCI Express® peripheral component interconnect express bus packet(s) are utilized. For memory transaction(s) (*e.g.,* read and/or write), a packet is formed that includes a requester ID (*e.g.,* source identifier), a transaction type (*e.g.,* memory read or memory write) and memory address(es). The requester ID (*e.g.,* source identifier) identifies the source of a memory transaction. The system can employ the requester ID (*e.g.,* source identifier) to identify a source of a disallowed DMA memory transaction.

Please amend the paragraph beginning on page 3, line 23 starting with "Another aspect of" with the following amended paragraph:

Another aspect of the present invention provides for a direct memory access memory corruption detection system having a memory controller having an access table. The memory controller is coupled to device(s) *via* a transaction-based bus (*e.g.,* PCI Express® peripheral component interconnect ~~interface~~ express bus). In one example, the system further includes a device driver, DMA API(S) and a memory manager. The device driver programs the device for a DMA read and/or write operation. The device driver further employs the DMA API(s) to allocate a region of physical memory for the device to use for DMA. The physical address is then programmed into the DMA engine for the device. When the device driver calls into the DMA API(s), the operating system programs information into the memory controller. For example, a range of physical memory, a source identifier and/or access information can be provided to the memory controller and stored in the access table.

Please amend the paragraph beginning on page 4, line 3 starting with "After programming this" with the following amended paragraph:

After programming this new "row" of information into the memory controller access table, the memory controller can monitor for memory transaction(s) in the particular range of memory and/or by the particular source (*e.g.,* identified by the source identifier). Thereafter, the device attempts to perform a DMA transaction and provides a request to the memory controller *via* the transaction based bus (*e.g.,* employing a PCI express® peripheral component interconnect express bus packet). The memory controller determines whether the requested DMA transaction is allowed. For example, the memory controller can determine if the address of the memory transaction is in one of the allowed ranges. If so, it determines whether a source of the requested DMA transaction is the same as the source identifier stored with the allowed range in the access table. Finally, the memory controller determines whether the requested DMA transaction type matches the access attribute(s) stored in the access table. If the condition(s) are met, the memory controller permits the requested DMA memory transaction to proceed. If the conditions are not met, the memory controller can reject the memory access along with, optionally, providing error information to the CPU. For example, error information could be provided to the operating system executing on the CPU (*e.g., via* a corrected platform error (CPE) event).

Please amend the paragraph beginning on page 6, line12 starting with "The system and" with the following amended paragraph:

The system and method of the present invention facilitate detection of direct memory access memory transaction(s) that can, if permitted, cause memory corruption. For purposes of explanation, the present invention will be described in the context of a PCI Express® peripheral component interconnect ~~interface~~ express bus. However, those skilled in the art will recognize that any suitable bus can be employed and all such types of buses are intended to fall within the scope of the hereto appended claims.

Please amend the paragraph beginning on page 6, line18 starting with "Referring to Fig. 1" with the following amended paragraph:

Referring to Fig. 1, a direct memory access memory corruption detection system 100 in accordance with an aspect of the present invention is illustrated. The system 100 can facilitate detection of direct memory access transaction(s) that can, if permitted, cause memory corruption. Historically, memory corruption is a major point of user dissatisfaction with computer system(s) and/or operating system(s). The system 100 can mitigate DMA memory corruption by employing transaction-based DMA bus system(s) (*e.g.,* PCI Express® peripheral component interconnect ~~interface~~ express). As noted previously, DMA transaction(s) occur outside the processor and cannot normally be traced, however the system 100 is extended to include an interface to specify "allowed" and/or "disallowed" memory range(s) for a DMA transaction. If a DMA transaction occurs in a disallowed range, then it is rejected and, optionally, error information is provided to the operating system running on the processor.

Please amend the paragraph beginning on page 7, line20 starting with "Briefly, the Peripheral" with the following amended paragraph:

Briefly, the Peripheral Component Interface (PCI) Express is a high performance, general purposes I/O interconnect defined for a wide variety of future computing and communication platforms. The PCI Express architecture includes three discrete logical layers: the transaction layer, the data link layer and the physical layer. PCI Express® peripheral component interconnect ~~interface~~ express uses packets to communicate information between components. Packets are formed in the transaction and data link layers to carry the information from the transmitting component to the receiving component. For memory transaction(s) (*e.g.,* read and/or write), a packet is formed that includes a requester ID (*e.g.,* source identifier), a transaction type (*e.g.,* memory read or memory write) and memory address(es). The requester ID (*e.g.,* source identifier) identifies the device which is the source of a memory transaction. The packet includes additional fields discussion of which is omitted for purposes of brevity. The PCI Express® peripheral component interconnect ~~interface~~ express bus includes a source identifier for memory transaction(s) that includes "bus device function" style identification.

Please amend the paragraph beginning on page 8, line 3 starting with "In this example" with the following amended paragraph:

In this example, the memory controller 230 is coupled to device(s) (not shown) *via* a PCI Express® peripheral component interconnect ~~interface~~ express bus 240. At system initialization, for example, the memory controller 230 receives access information regarding device(s) which is stored in the access table 220.

Please amend the paragraph beginning on page 8, line 7 starting with "Referring briefly to" with the following amended paragraph:

Referring briefly to Fig. 3, an exemplary access table 300 in accordance with an aspect of the present invention is illustrated. The access table 300 includes a source identifier field 310, a memory address(es) field 320 and a access attribute(s) field 330. The source identifier field 310 includes unique identifying information regarding potential requester(s) of DMA transaction(s) (*e.g.,* device(s)). For example, in a PCI Express® peripheral component interconnect ~~interface~~ express system, the source identifier field would correspond to the source address in a PCI Express® peripheral component interconnect ~~interface~~ express packet.

Please amend the paragraph beginning on page 9, line 1 starting with "Next, turning to" with the following amended paragraph:

Next, turning to Fig. 4, a direct memory access memory corruption detection system 400 in accordance with an aspect of the present invention is illustrated. The system 400 includes a memory controller 210 having an access table 220, the memory controller facilitating access to memory 230. The memory controller 230 is coupled to a device 250 *via* a transaction-based bus (*e.g.,* PCI Express® peripheral component interconnect ~~interface~~ express bus) 240. The system 400 further includes a device driver 260, direct memory access operating system application programming interface(s) (DMA API(s)) 270 and/or a memory manager 280. As illustrated in Fig. 5, the system 400 can further include, optionally, an operating system CPE handler 290.

Please amend the paragraph beginning on page 9, line 29 starting with "The device driver" with the following amended paragraph:

The device driver 260 further programs the device 250 for a DMA read or write operation. Once the access table 220 has been programmed, at some point thereafter, the device 250 attempts to perform a DMA transaction. The device 250 provides a request to the memory controller 210 *via* the transaction based bus 240 (*e.g.,* employing a PCI express® peripheral component interconnect ~~interface~~ express bus packet). In one example, the memory controller 210 determines whether the requested DMA transaction is in one of the allowed ranges. If so, it determines whether a source of the requested DMA transaction is the same as the source id stored with the allowed range in the access table 220. Finally, the memory controller 210 determines whether the requested DMA transaction type matches the access attribute(s) stored in the access table 220. In another example, the memory controller 210 checks to see if there are any entry(ies) for the specified source identifier before looking for allowed/disallowed transaction(s).

Please amend the paragraph beginning on page 11, line 8 starting with "The PCI Express" with the following amended paragraph:

The PCI Express® peripheral component interconnect ~~interface~~ express bus includes a source identifier for memory transaction(s) that includes "bus device function" style identification. With this, fairly detailed information regarding a potential source of memory corruption can be identified. As noted previously, optionally, the memory controller 210 can provide error information *via* a CPE event to the operating system CPE handler 290.

Please amend the paragraph beginning on page 13, line 3 starting with "The system bus" with the following amended paragraph:

The system bus 718 can be any of several types of bus structure(s) including the memory bus or memory controller, a peripheral bus or external bus, and/or a local bus using any variety of available bus architectures including, but not limited to, an 8-bit bus, Industrial Standard

Architecture (ISA), Micro-Channel Architecture (MSA), Extended ISA (EISA), Intelligent Drive Electronics (IDE), VESA Local Bus (VLB), Peripheral Component Interconnect (PCI), PCI Express® peripheral component interconnect ~~interface~~ express, remote DMA (RDMA), Universal Serial Bus (USB), Advanced Graphics Port (AGP), Personal Computer Memory Card International Association bus (PCMCIA), and Small Computer Systems Interface (SCSI).

Please amend the paragraph beginning on page 20, line 2 starting with "A system and" with the following amended paragraph:

A system and method that facilitates detection of direct memory access (DMA) corruption is provided. The system can mitigate DMA memory corruption in computer system(s) employing transaction-based DMA bus system(s) (*e.g.,* PCI Express® peripheral component interconnect ~~interface~~ express).